10/511187

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

274, I-10148 Torino (IT). **MINGOZZI, Enzo** [IT/IT]; c/o Telecom Italia Lab S.p.a., Via Reiss Romoli, 274, I-10148 Torino (IT). **SCARRONE, Enzo** [IT/IT]; c/o Telecom Italia Lab S.p.a, Via Reiss Romoli, 274, I-10148 Torino (IT). **STEA, Giovanni** [IT/IT]; c/o Telecom Italia Lab S.p.a., Via Reiss Romoli, 274, I-10148 Torino (IT).
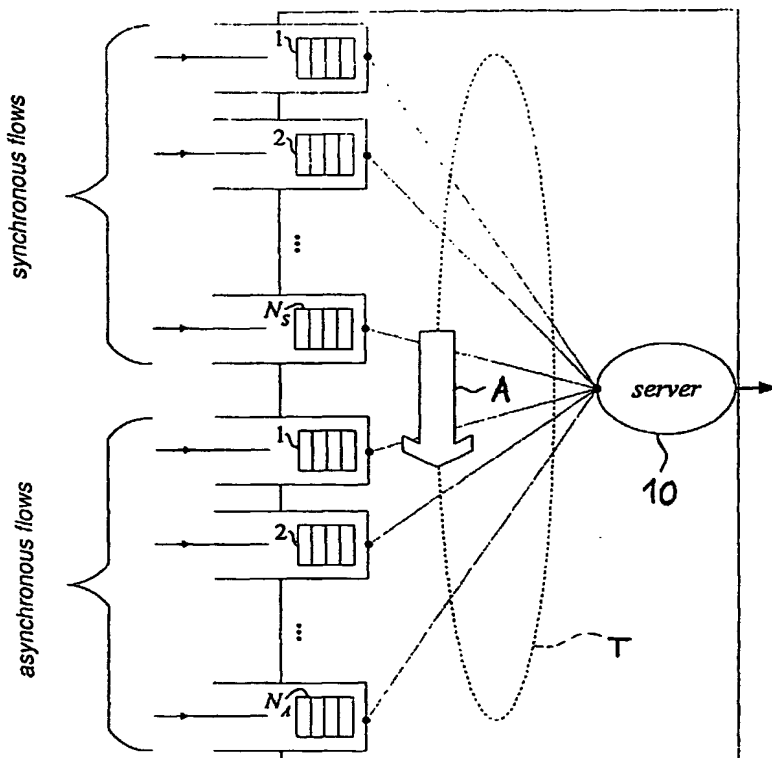
(74) Agents: GIANNESI, Pier Giovanni et al.; Pirelli S.P.A., Viale Sarca, 222, I-20126 Milano (IT).

---

(54) Title: SCHEDULING A SHARED RESOURCE AMONG SYNCHRONOUS AND ASYNCHRONOUS PACKET FLOWS

(57) Abstract: Each synchronous flow ($i=1, 2, ..., N_s$) is associated to a respective synchronous capacity value (H_i) that is related the period of time for which a synchronous flow can be serviced before the server moves on. This value can be selected either according to a local allocation criteria or according to a global allocation criteria. Each asynchronous flow ($i=1, 2, ..., N_A$) s is associated to a respective first value indicating the delay to be made up so that the respective queue has the right to be serviced and to another value indicating the instant in which the server visited the respective queue in the previous cycle. Each queue associated to a synchronous flow (h) is then serviced for a period of time that is related to be aforesaid synchronous capacity value, while each queue associated to an asynchronous flow (i) is serviced only if the server's visit occurs before the expected moment. The server's visit (10) to the synchronous queues should preferably take place during two successive cycles in order to optimise the use of the resources available.

European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii)) for the following designations AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZM, ZW, ARIPO patent (GH, GM, KE, LS,*

MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG)

— *of inventorship (Rule 4.17(iv)) for US only*

Published:

— *with international search report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

SCHEDULING A SHARED RESOURCE AMONG SYNCHRONOUS AND ASYNCHRONOUS PACKET FLOWS

TECHNIQUE SECTOR

This invention refers to the packet communication
5   systems, and in particular to the scheduling criteria of a
shared resource, i.e. the criteria used to select the packet
to which the resource is to be assigned each time this
occurs.

The solution given in the invention has been developed
10   both for radio resource scheduling (e.g.: MAC or Medium
Access Control level scheduling), and for the scheduling of
computational and transmissive resources in the network
nodes, for example, for flow scheduling with different
service quality on Internet Protocol router (IP). The
15   following description is based especially on the latter
application example, and is given purely as an example and
does not limit the scope of the invention.

INTRODUCTION

For several years now, the widespread application and
20   rapid evolution of the packet networks have given rise to the
problem of integrating the traditional services offered by
the old generation packet networks (electronic mail, web
surfing, etc.) and the new services previously reserved for
circuit switching networks (real-time video, telephony, etc.)
25   into the so-called integrated services networks.

Systems like UMTS, for example, for which a fixed packet
network component (core network) is envisaged, must
simultaneously handle voice and data services, and offer
support for the development of new services be they real-time
30   or not.

The integrated services networks must therefore be able
to handle traffic flows with different characteristics and to
offer each type of flow a suitable service quality, a set of
performance indexes negotiated between user and service

provider, which must be guaranteed within the terms agreed upon.

One of the key elements in providing the service quality requested is the scheduling system implemented on the network
5   nodes, i.e. the system used to select the packet to be transmitted from those present on the node; this system must obviously embody contrasting characteristics like flexibility, in terms of capacity to provide different types of services, simplicity, a characteristic that makes it
10   possible to use in environments that require high transmission speeds and the handling of numerous transmission flows, and efficiency in the use of the shared resource (e.g. the transmissive means).

The need to guarantee a given level of service quality
15   (or Qos) in the packet networks is constantly increasing, as can be seen for example in the documents US-A-6 091 709, US-A-6 147 970 or EP-A-1 035 751.

This invention in fact is the development of the solution described in the industrial invention patent request
20   TO2000A001000 and in the corresponding request PCT/IT 01/00536.

The previous solution basically applies to the scheduling of a service resource shared between several information packet flows in which the flows generate
25   respective associated queues and are serviced when the server gives permission to transmit.

The flows are divided into synchronous flows, which require a minimum service rate guarantee, and into asynchronous flows, which use the service capacity of the
30   resource that is left unused by the synchronous flows. The solution in question includes the following:

- provides a server that visits the queues associated with the flows in successive cycles, granting each queue a target token rotation time (or "revolution"), called TTRT,

which identifies the time required for the server to complete the queue visiting cycle,

- associates each synchronous flow with a synchronous capacity value indicating the maximum time the synchronous

5   flow can be serviced before its transmission permission is revoked by the server,

- associates each asynchronous flow with a first lateness(i) value, indicating the delay that must be made up for the respective queue to have the right to be serviced,

10   plus another value (last_token_time) indicating the moment the server visited the respective queue in the previous cycle, which determines the time elapsed since the server's previous visit,

- services each queue associated to a synchronous flow

15   for a maximum period of time equal to the above-mentioned synchronous capacity value, and

- services each queue associated to an asynchronous flow only if the server's visit occurs before the expected moment. This advance is obtained from the difference between the

20   aforesaid TTRT time and the time that has elapsed since the server's previous visit and the accumulated delay.

If this difference is positive it defines the maximum service time for each queue associated to an asynchronous flow.

25   The solution referred to above has proved to be completely satisfactory from an operational point of view. The experience gained by the "Petitioner" has however shown that the solution can be further developed and improved as illustrated in this invention.

30   This applies particularly to the following aspects:

- the possibility of offering different types of service while keeping computational costs low: an important feature for computer network applications that must guarantee service quality for its users, like the IP networks with Intserv

(Integrated Services, as per IETF specification) or Diffserv (Differentiated Integrated Services, as per IETF specification), or for the radio resource scheduling systems like the MAC level scheduling algorithms (W-LAN systems,

5   third generation radio-mobile services);

- the possibility of guaranteeing the bit rate of the various flows, the maximum queuing delay and the maximum occupation of the buffers of each flow for synchronous traffic;

10   - flexibility, in terms of capacity to provide two different types of services at the same time, rate-guaranteed (suitable for synchronous flows) and fair queuing (suitable for asynchronous flows), especially in service integration networks;

15   - the possibility of isolating transmission flows, i.e. it makes the service offered to a single flow independent from the presence and behaviour of other flows;

- low computational complexity in terms of the number of operations necessary to select the packet to be transmitted;

20   this feature makes it possible to use in environments that require high transmission speeds and the handling of numerous transmission flows, also in view of a possible implementation in hardware;

- adaptability, in the sense that it can handle a change

25   in the operating parameters (e.g. the number of flows present) by redistributing its resources without having to resort to complex procedures; and

- analytic describability, i.e. it gives a complete analytic description of the system's behaviour, which makes

30   it possible to relate the service quality measurements to the system parameters.

Another important aspect is equity, i.e. the possibility to manage in the same way both the transmission flows that receive a rate-guaranteed service, and those that receive a

fair-queuing service, giving each one a level of service that is proportional to that requested, even in the presence of packets of different lengths.

DESCRIPTION OF THE INVENTION

5   The aim of this invention is to develop even further the already known solution referred to previously with special attention to the aforesaid aspects.

According to this invention, this aim can be reached by using a scheduling procedure having the characteristics
10   referred to specifically in the following claims.

The invention also refers to the relative system.

Briefly, the solution given in the invention operates a scheduling system that can be defined with the name introduced in this patent request - Packet Timed Token
15   Service Discipline or PTTSD.

At the moment, this scheduling system is designed to work on a packet-computer network switching node and is able to multiplex a single transmission channel into several transmission flows.

20   The system offers two different types of service: rate-guaranteed service, suitable for transmission flows (henceforth, "synchronous flows") that require a guaranteed minimum service rate, and a fair-queueing service, suitable for transmission flows (henceforth "asynchronous flows") that
25   do not require any guarantee on the minimum service rate, but which benefit from the greater transmission capacity available. The system provides the latter, however, with an equal sharing of the transmission capacity not used by the synchronous flows.

30   The traffic from each transmission flow input on the node is inserted in its own queue (synchronous or asynchronous queues) from which it will be taken to be transmitted. The server visits the queues in a fixed cyclic

order and grants each queue a service time established according to precise timing constraints at each visit.

5   The server initially visits the synchronous queues twice during a revolution, thus completing a major cycle and a minor or recovery cycle, and then moves on to visit the asynchronous queues.

BRIEF DESCRIPTION OF THE FIGURE

The following description of the invention is given as a non-limiting example, with reference to the annexed drawing,
10   which includes a single block diagram figure that illustrates the operating criteria of a system working according to the invention.

DESCRIPTION OF A PREFERRED FORM OF EXECUTION

A scheduling system as given in the invention is able to
15   multiplex a single transmission channel into several transmission flows.

The system offers two different types of service: a rate-guaranteed service, suitable for transmission flows (henceforth i synchronous flows where i = 1, 2, ..., N_S) that
20   require a guaranteed minimum service rate, and a best-effort service, suitable for transmission flows (henceforth j asynchronous flows where j = 1, 2, ..., N_A) that do not require any guarantee on the service rate. The system provides the latter, however, with an equal sharing of the transmission
25   capacity not used by the synchronous flows.

It should be supposed that N_S and N_A are non-negative integers and that each synchronous flow $i = 1..N_s$ requires a service rate equal to r_i , and that the sum of the service rates requested by the synchronous flow does not exceed the
30   capacity of channel C ($\sum_{i=1}^{N_S} r_i \leq C$) .

The traffic from each transmission flow input on the node is inserted in its own queue (synchronous or asynchronous queues will be discussed later) from which it

will be taken to be transmitted. The server 10 visits the queues in a fixed cyclic order (ideally illustrated in the figure of the drawings with trajectory T and arrow A), granting each queue a service time established according to precise timing constraints at each visit.

The procedure referred to in the invention includes an initialisation stage followed by cyclic visits to the queues. These procedures will be discussed below.

Initialisation

First of all, it is necessary to give the system the information relating to the working conditions: how many synchronous flows there are (in general: N_S), what the transmission rate requested by each synchronous flow is, how many asynchronous flows there are, the target rotation time (TTRT), i.e. how long a complete cycle during which the sever visits all the queues once is to last.

Synchronous flows

Each synchronous flow i, $i = 1..N_s$, is associated, according to an appropriate allocation policy, to a variable H_i (synchronous capacity), which measures the maximum time for which the traffic of a synchronous flow can be transmitted before the server takes the transmission permission away. The possible allocation policies will be described below. A variable Δ_i, initially nil, is associated to each synchronous flow, and stores the amount of transmission time available to the flow.

Asynchronous flows

Each asynchronous flow j, $j = 1..N_A$, is associated with two variables, L_j and $last\_visit\_time_j$; the first variable stores the delay or lag that must be made up for the asynchronous queue j to have the right to be serviced; the second variable stores the instant the server visited the

asynchronous queue j in the previous cycle. These variables are respectively initialised to zero and to the instant the revolution in progress when the flow is activated started.

This way of proceeding means that the asynchronous flows can be activated at any moment, not necessarily at system startup.

<u>Visit to a generic synchronous queue i, with i = 1...N_S during the major cycle</u>

A synchronous queue can be serviced for a period of time equal to the maximum value of the variable Δ_i. This variable is incremented by H_i (value decided during initialisation) when the queue is visited in the major cycle, and decremented by the transmission time of each packet transmitted.

The service of a queue during the major cycle ends when either the queue is empty (in which case the variable Δ_i is reset), or the time available (represented by the current value of Δ_i) is not sufficient to transmit the packet that is at the front of the queue.

<u>Visit to a generic synchronous queue i, i = 1...N_S during the minor cycle</u>

During the minor (or recovery) cycle a synchronous queue can transmit only one packet, provided the variable Δ_i has a strictly positive value. If transmission takes place, the variable Δ_i is decremented by the transmission time.

<u>Visit to a generic asynchronous queue j, with j =1,...,N_A</u>

An asynchronous queue can only be serviced if the server's visit takes place before the expected instant. To calculate whether the server's visit is in advance, subtract the time that has elapsed since the previous visit and the accumulated delay L_j from the target rotation time $TTRT$.

If this difference is positive, it is the period of time for which the asynchronous queue j has the right to be serviced, and in this case the variable L_j is reset.

If the difference is negative, the server is late and the queue j cannot be serviced; in this case the delay is stored in the variable L_j. The asynchronous queue service ends when the queue is empty, or the time available (which is decremented each time a packet is transmitted) is not sufficient to transmit the packet that is at the front of the queue.

Visit sequence during a revolution

A double scan is made on all the synchronous queues (major and minor cycles) during one revolution, and then the asynchronous queues are visited. The minor cycle ends the moment one of the following events takes place:

- the last synchronous queue has been visited;

- a period of time that is equal to or greater than the sum of the capacity of all the synchronous queues has elapsed since the beginning of the major cycle.

Analytic guarantees

The synchronous capacities are linked to the target rotation time $TTRT$ and to the duration of the transmission of the longest packet τ_{max} by the following inequality, which must always be verified:

$$\sum_{i=1}^{N_s} H_i + \tau_{max} \leq TTRT \quad (1)$$

Minimum transmission rate for synchronous flows

In hypothesis (1), the system as illustrated herein guarantees that the following normalised transmission rate will be guaranteed for each synchronous flow:

$$\gamma_i = \frac{N_A + 1}{N_A + \sum_{h=1}^{N_s} X_h + \alpha} \cdot X_i$$

with:

9

$$X_i = H_i/TTRT$$

$$\alpha = \tau_{max}/TTRT$$

and it is also possible to guarantee that, given any period of time $[t_1, t_2)$ in which the generic synchronous queue i is never empty, the service time $W_i(t_1, t_2)$ received from the queue i in $[t_1, t_2)$ verifies the following inequality:

$$\gamma_i \cdot (t_2 - t_1) - W_i(t_1, t_2) \leq \Lambda_i < \infty \quad (2)$$

where:

$$\Lambda_i = \begin{cases} H_i \cdot (2 - \gamma_i) + (1 + \gamma_i) \cdot \tau_i & se \ H_i \geq \tau_i \\ \tau_i + 2 \cdot H_i & se \ H_i < \tau_i \end{cases}$$

and τ_i is the transmission time of the longest packet for the flow i.

Expression (2) seen previously establishes that the service supplied by the i synchronous flow system of the type described here does not differ by more than Λ_i from the service that the same flow would experience if it were the only owner of a private transmission channel with a capacity equal to γ_i times that of the channel managed by the system illustrated in this invention. Λ_i therefore represents the maximum service difference with respect to an ideal situation.

A synchronous flow can therefore feature a parameter, called latency, which is calculated as follows:

$$\Theta_i = \begin{cases} \left(2 + \dfrac{\tau_i}{H_i}\right) \dfrac{N_A TTRT + \tau_{max} + \sum\limits_{h \in S} H_i}{N_A + 1} + \tau_i - H_i, & se \ H_i \geq \tau_i \\[4ex] \left(2 + \dfrac{\tau_i}{H_i}\right) \dfrac{N_A TTRT + \tau_{max} + \sum\limits_{h \in S} H_i}{N_A + 1}, & se \ H_i < \tau_i \end{cases}$$

or, for $N_A \to \infty$:

10

$$\Theta_i^* = \begin{cases} \left(2 + \dfrac{\tau_i}{H_i}\right)TTRT + \tau_i - H_i, & \text{se } H_i \geq \tau_i \\ \left(2 + \dfrac{\tau_i}{H_i}\right)TTRT, & \text{se } H_i < \tau_i \end{cases}$$

Given a switching node that implements the solution described herein, if the traffic input on a synchronous flow on that node is limited by a so-called "*leaky-bucket*" of parameters (σ, ρ), the following guarantees can be given:

a) Maximum delay on a single node for a synchronous flow

Each packet has a delay that is not greater than:

$$D = \sigma/\rho + \Theta_i$$

b) Maximum memory occupation on a node for a synchronous flow

The amount of memory occupied by packets in a synchronous flow packet is:

$$B = \sigma + \rho ? \Theta_i$$

c) Maximum delay on a route of N nodes for a synchronous flow

Let $\Phi_1 ... \Phi_N$ N be switching nodes that implement the system described herein; let Θ_i^j be the latencies calculated on each of the Φ_j nodes and let:

$$\overline{\Theta}_i = \sum_{j=1}^{N} \Theta_i^j$$

In this case it is possible to define an upper limit for the maximum delay for a packet to cross the N nodes, provided that the traffic input on the first node is limited by a *leaky-bucket* of parameters (σ, ρ); this limit is:

$$D_N = \sigma/\rho + \overline{\Theta}_i$$

The value $\Theta_i^* \geq \Theta_i$ can be employed in each of the three guarantees a), b), c); this means that the limits that do not depend on the number of active asynchronous flows can be calculated.

Parameter selection

The ability to guarantee that the synchronous flows receive a minimum service rate no lower than that requested is subordinate to a correct selection of the synchronous capacities H_i, $i=1..N_s$. Assuming that each synchronous flow i requires a minimum transmission rate r_i, it is necessary to allocate the synchronous capacities to verify the following inequality:

$$\gamma_i \geq r_i / C \quad (3)$$

The solution described herein allocates the synchronous capacities according to two different schemes called local and global allocation respectively.

Local allocation

The synchronous capacities are selected as follows :

$$H_i = \frac{r_i \cdot TTRT}{C}$$

In this way, the inequality (1) is verified if the transmission rates requested verify the following inequality:

$$\sum_{h=1}^{N_s} r_h / C \leq 1 - \alpha \quad (4)$$

Each synchronous flow is guaranteed a normalised service rate equal to:

$$\gamma_i = \frac{[N_A + 1] \cdot r_i / C}{N_A + \sum_{h=1}^{N_s} r_h / C + \alpha} \quad (5)$$

The value of γ_i given by expression (5) verifies the inequality (3).

Global allocation

According to this scheme, which requires $N_A > 0$, the synchronous capacities are selected as follows:

$$H_i = \frac{(N_A + \alpha) \cdot r_i / C}{N_A + 1 - \sum_{h=1}^{N_s} r_h / C} \cdot TTRT$$

In the global allocation scheme the sum of the transmission rates requested must also remain below the inequality (4). If (4) is verified, the normalised service rate of a synchronised flow is $\gamma_i = r_i/C$.

5   The global scheme guarantees greater use of the channel's transmission capacity than the local scheme, in that it allocates less capacity to the synchronous flows, leaving more bandwidth for the asynchronous flow transmission.

10   On the other hand, the use of a global scheme means that all the synchronous capacities are to be recalculated each time the number of flows (synchronous or asynchronous) present in the system changes; the use of a local scheme, however, means that the capacities can be established

15   independently from the number of flows in the system.

Selection of *TTRT*

The following scheme can be given to show the selection of *TTRT* in the solution according to the invention.

Given a set of synchronous flows with requested

20   transmission rates that verify the inequality:

$$\sum\nolimits_{h=1}^{N_S} r_h/C < 1$$

TTRT must be selected according to the following inequality:

$$TTRT \geq \frac{\tau_{max}}{1 - \sum\nolimits_{h=1}^{N_S} r_h/C}$$

25   The pseudo-code illustrated below analytically describes the behaviour of a system as given in the invention.

Flow initialisation

```
Sync_Flow_Init (synchronous flow i)
30    {    Δi=0;
          Select_synchronous_bandwidth Hi;
      }
```

```
Async_Flow_Init (asynchronous flow j)
{      Lⱼ = 0 ;
       last_visit_timeⱼ = start_of_curr_revolution;
}
```

Visit to a generic synchronous queue i, $i = 1 \ldots N_S$, during the major cycle

```
Major_Cycle_Visit (synchronous flow i)
{      Δᵢ+=  Hᵢ;
       q=first_packet_transmission_time;
       while ((Δᵢ>=q)  and (q > 0))
       {      transmit_packet (q);
              Δᵢ -= q;
              elapsed_time+= q;
       }
       if (q=0) Δᵢ=0;
```

Visit to a generic synchronous queue i, $i = 1 \ldots N_S$, during the minor cycle

```
Minor_Cycle_Visit (synchronous flow i)
{      q=first_packet_transmission_time;
       if (q > 0)
       {
              transmit_packet (q);
              Δᵢ -= q;
              elapsed_time += q;
       }
       if (q=0) Δᵢ=0;
}
```

Visit to a generic asynchronous queue j, $j =1 \ldots N_A$

```
Async_Flow_Visit (asynchronous flow j)
{      t = current_time;
       earliness = TTRT-Lⱼ — (t-last_visit_timeⱼ);
       if ( earliness > 0 )
       {      Lⱼ = 0;
              transmit_time = earliness;
              q=first_packet_transmission_time;
              while ((transmit_time>=q) and (q > 0))
              {
                     transmit_packet (q);
                     transmit_time -= q;
              }
       }
       else Lⱼ = - earliness;
```

```
          last_visit_time_j = t;
     }
```

Visit sequence during a revolution

```
5    PTTSD revolution ()
     {     elapsed_time=0;
           for (i=1 to N_S) Major_Cycle_Visit (i);
           i = 1;
           while((elapsed_time<sum(H_n)) and (i<=N_S))
10         {
                 if (Δ_i>0) Minor_Cycle_Visit (i);
                 i ++;
           }
           for (j=1 to N_A) Async_Flow_Visit (j);
15   }
```

Obviously the details of how this is done can be altered with respect to what has been described, without however, leaving the context of this invention.

20

CLAIMS

1. Procedure for the scheduling of a service resource shared among several information packet flows that generate respective associated queues, said flows including

5 synchronous flows (i = 1, 2, ..., N_S) that require a guaranteed minimum service rate (r_i) and asynchronous flows (j = 1, 2, ..., N_A) that use the service capacity of said resource that is left unused by the synchronous flows, the procedure making use of a server (10) and comprising the

10 following operations:

- makes said server (10) visit the respective queues associated to said flows (i, j) in successive cycles on the basis of the target rotation time value (TTRT), which identifies the time necessary for the server (10) to complete

15 a visit cycle on said respective queues,

- associates each synchronous flow (i) with a respective synchronous capacity value (H_i) indicating the maximum period of time for which the respective synchronous flow can be serviced before the server moves on,

20 - associates each asynchronous flow (j) with a first respective delay value (L_j) that identifies the value that must be made up for the respective queue to have the right to be serviced, and a second respective value (*last_visit_time*) that indicates the instant in which the server (10) visited

25 the respective queue in the previous cycle, determining for said respective queue, the time that has elapsed since the server's previous visit,

- services each queue associated to a synchronous flow (i) for a maximum service time relative to said respective

30 value of synchronous capacity (H_i), and

- services each queue associated to an asynchronous flow (j) only if the server's visit (10) occurs before the expected instant, said advance being determined as the difference between said target rotation time value (TTRT) and

the time that has elapsed since the server's (10) previous
visit and the accumulated delay; if positive, this difference
defines the maximum service time for each asynchronous queue,

the procedure also includes the operation that defines
said respective synchronous capacity value (H_i) for the queue
associated to the i-th synchronous flow by satisfying:

- i) the expressions

$$\sum_{i=1}^{N_s} H_i + \tau_{max} \le TTRT$$

$$TTRT \ge \frac{\tau_{max}}{1 - \sum_{h=1}^{N_s} r_h / C}$$

- ii) as well as at least one of the following expressions

$$H_i = \frac{r_i \cdot TTRT}{C} \quad \text{and}$$

$$H_i = \frac{(N_A + \alpha) \cdot r_i / C}{N_A + 1 - \sum_{h=1}^{N_s} r_h / C} \cdot TTRT$$

where:

- H_i is said respective synchronous capacity value (H_i)
  for the queue associated to the i-th synchronous flow,
- the summations are extended to all the synchronous
  flows, equal to N_s,
- N_A is the number of said asynchronous flows,
- τ_{max} is the duration of the longest packet service by
  said shared service resource,
- TTRT is said target rotation time value,
- C is the service capacity of said shared service
  resource,
- r_i is the minimum service rate required by the i-th
  synchronous flow, with $\sum_{h=1}^{N_s} r_h / C < 1$, and
- α is a parameter that gives $\sum_{h=1}^{N_s} r_h / C \le 1 - \alpha$.

2. Procedure as per claim 1, characterised by the fact
that during each of said successive cycles, said server (10)

17

performs a double scan on all the queues associated to said synchronous flows (i = 1, 2, ..., N_S) and then visits the queues associated to said asynchronous flows (j = 1, 2, ..., N_A).

5   **3.** Procedure as per claim 2, characterised by the fact that it includes the following operations:

- associates to each synchronous flow (i) a further value (Δ_i) indicating the amount of service time that is available to the respective flow,

10   - during a major cycle of the said double scan it services each queue associated to a synchronous flow (i) for a period of time equal to the maximum said further value (Δ_i), and

- during a minor cycle of said double scan it services
15   only one packet of each queue associated to a synchronous flow (i), provided that said further value (Δ_i) is strictly positive.

4. Procedure as per claim 3, characterised by the fact that it includes the operation of incrementing said further
20   value (Δ_i) by said respective value of the synchronous capacity (H_i) when the queue is visited during the major cycle of said double scan.

5. Procedure as per claim 3 or claim 4, characterised by the fact that it includes the operation of decrementing said
25   further value (Δ_i) of the transmission time by each packet serviced.

6. Procedure as per any of the claims 3 to 5, characterised by the fact that the service of each queue associated to a synchronous flow (i) during the major cycle
30   of said double scan ends when one of the following conditions occurs:

- the queue is empty,

- the time available, represented by said further value (Δ_i), is not sufficient to service the packet at the front of the queue.

7. Procedure as per claim 6, <u>characterised by the fact</u> that it includes the operation of resetting said further value (Δ_i) when the respective queue is empty.

8. Procedure as per any of the claims 3 to 7, <u>characterised by the fact</u> that it includes the operation of decrementing the service time of said further value (Δ_i) in the presence of a service given during the minor cycle of said double scan.

9. Procedure as per any of the claims 3 to 8, <u>characterised by the fact</u> that during said double scan of all the queues associated to said synchronous flows (i), said minor cycle ends when one of the following conditions is satisfied:

- the last queue associated to a synchronous flow (i) has been visited,

- a period of time not less that the sum of the capacities (Hi) of all the queues associated to said synchronous flows (i) has elapsed since the beginning of said major cycle of said double scan.

10. Procedure as per any of the claims 3 to 9, <u>characterised by the fact</u> that it includes the operation of initialising said further value (Δ_i) to zero.

11. Procedure as per any of the previous claims, <u>characterised by the fact</u> that in the case that said difference is negative, each said queue associated to an asynchronous flow (j) is not serviced and the value of said difference is accumulated with said delay (L_j).

12. Procedure as per any of the claims 1 to 11, <u>characterised by the fact</u> that the service of a queue associated to an asynchronous flow (j) ends when one of the following conditions is satisfied:

- the queue is empty,

- the time available is not sufficient to transmit the packet that is at the front of the queue.

13. Procedure as per any of the claims 1 to 12, <u>characterised by the fact</u> that said first respective value (L_j) and said second respective value (*last_visit_time*) are respectively initialised to zero and to the moment of startup of the current cycle when the flow is activated.

14. System for the scheduling of a service resource shared among several information packets that generate respective associated queues. Said flows include synchronous flows (i = 1, 2, ..., N_S) that require a guaranteed minimum service rate and asynchronous flows (j = 1, 2, ..., N_A) destined to use the service capacity of said resource left unused by the synchronous flows. The system also includes a server (10) able to visit the respective queues associated to said flows (i, j) in successive cycles, which is configured to perform the following operations:

- determine a target rotation time value (TTRT) that identifies the time necessary for the server (10) to complete a visiting cycle of said respective queues,

- associate to each synchronous flow (i) a respective synchronous capacity value (H_i) indicating the maximum amount of time for which a synchronous flow can be serviced before moving on to the next,

- associate to each asynchronous flow (j) a first respective delay value (L_j) that identifies the delay that must be made up for the respective queue to have the right to be serviced, and a second respective value (*last_visit_time*) that indicates the instant in which in the previous cycle the server (10) visited the respective queue, determining for said respective queue, the time that has elapsed since the server's (10) previous visit,

- service each queue associated to a synchronous flow (i) for a maximum period of time relating to said respective synchronous capacity value (H_i), and

- service each queue associated to an asynchronous flow (j) only if the server's visit (10) occurs before the expected instant, said advance being determined as the difference between said target rotation time (TTRT) and the time that has elapsed since the server's (10) previous visit and the accumulated delay; if positive, this difference defines the maximum service time for each said asynchronous queue.

the system is configured to define said respective synchronous capacity value (H_i) for the queue associated to the i-th synchronous flow so that the following are satisfied:

- i) the expressions

$$\sum_{i=1}^{N_s} H_i + \tau_{\max} \leq TTRT$$

$$TTRT \geq \frac{\tau_{\max}}{1 - \sum_{h=1}^{N_s} r_h / C}$$

- ii) as well as at least one of the following expressions

$$H_i = \frac{r_i \cdot TTRT}{C} \quad \text{and}$$

$$H_i = \frac{(N_A + \alpha) \cdot r_i / C}{N_A + 1 - \sum_{h=1}^{N_s} r_h / C} \cdot TTRT$$

where:

- H_i is the said respective synchronous capacity value (H_i) for the queue associated to the i-th synchronous flow,

- the summations are extended to all the synchronous flows, equal to N_s,

- N_A is the number of said asynchronous flows,

21

.- τ_{max} is the service duration of the longest packet by
   said shared service resource,

- TTRT is said target rotation time value,

- C is the service capacity of said shared service
   resource,

- r_i is the minimum service rate requested by the i-th

   synchronous flow, with $\sum_{h=1}^{N_s} r_h/C < 1$, and

- α is a parameter that gives $\sum_{h=1}^{N_s} r_h/C \le 1-\alpha$.

   15. System as per claim 14, <u>characterised by the fact</u>
that during each of the said successive cycles, said server
(10) performs a double scan on all the queues associated to
said synchronous flow (i = 1, 2, ..., N_S) and then visits the
queues associated to said asynchronous flows (j = 1, 2, ...,
N_A).

   16. System as per claim 15, <u>characterised by the fact</u>
that:

- a further value (Δ_i), indicating the amount of service
time available to the respective flow, is associated to each
synchronous flow (i),

- during a major cycle of said double scan, each queue
associated to a synchronous flow (i) is serviced for a period
of time equal to the maximum further value (Δ_i), and

- during a minor cycle of said double scan the system
services only one packet of each queue associated to a
synchronised flow (i), provided said further value (Δ_i) is
strictly positive.

   17. System as per claim 16, <u>characterised by the fact</u>
that said further value (Δ_i) is incremented by said
respective synchronous capacity value (H_i) when the queue is
visited during the major double scan cycle.

   18. System as per claim 16 or claim 17, <u>characterised by</u>
<u>the fact</u> that said further value (Δ_i) is decremented by the
transmission time of each packet serviced.

19. System as per any of the claims 16 to 18, <u>characterised by the fact</u> that the system is configured so that the service of each queue associated to a synchronous flow (i) during the major cycle of said double scan ends when one of the following conditions occurs:

- the queue is empty,

- the time available, represented by said further value (Δ_i), is not sufficient to serve the packet at the front of the queue.

20. System as per claim 19, <u>characterised by the fact</u> that said further value (Δ_i) is reset when the respective queue is empty.

21. System as per any of the claims 16 to 20, <u>characterised by the fact</u> that in the presence of a service given during the minor cycle of said double scan, said further value (Δ_i) is decremented by the amount of service time.

22. System as per any of the claims 16 to 21, <u>characterised by the fact</u> that during said double scan on all the queues associated to said synchronous flows (i), said minor cycle ends when one of the following conditions is satisfied:

- the last queue associated to a synchronous flow (i) has been visited,

- a period of time not less than the sum of the capacities (Hi) of all the queues associated to said synchronous flows (i) has elapsed since the beginning of said major cycle of said double scan.

23. System as per any of the previous claims 16 to 22, <u>characterised by the fact</u> that said further value (Δ_i) is initialised to zero.

24. System as per any of the previous claims 16 to 23, <u>characterised by the fact</u> that if said difference is negative, each said queue associated to an asynchronous flow

(j) is not serviced and the value of said difference is accumulated with said delay (L_j).

25. System as per any of the claims 14 to 24, characterised by the fact that the service of a queue associated to an asynchronous flow (j) ends when one of the following conditions is satisfied:

- the queue is empty,

- the time available is not sufficient to transmit the packet that is at the front of the queue.

26. System as per any of the claims 14 to 25, characterised by the fact that said first respective value (L_j) and said second respective value (*last_visit_time*) are respectively initialised to zero and to the moment of startup of the current cycle when the flow is activated.
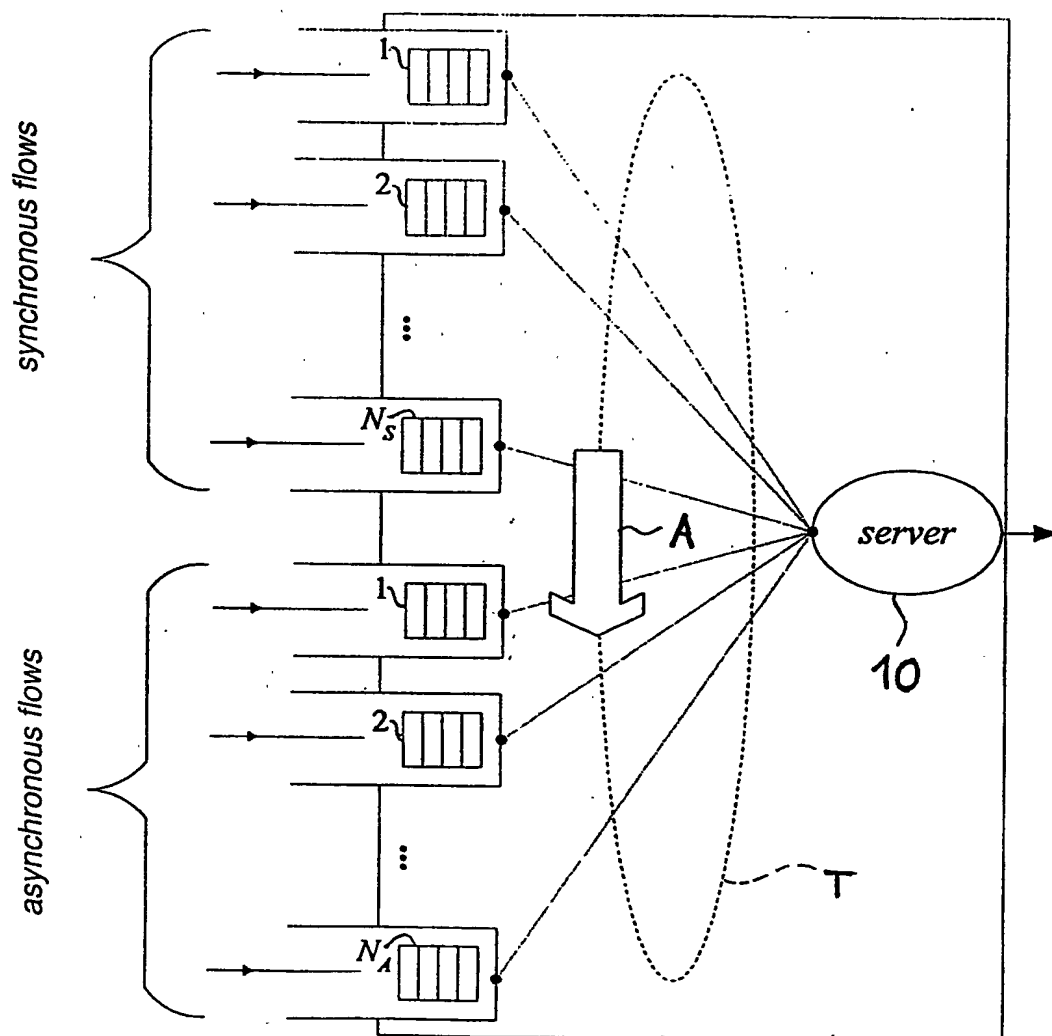
INTERNATIONAL SEARCH REPORT

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7   H04L12/64   H04L12/56

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7   H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	HAN C-C ET AL: "A POLYNOMIAL-TIME OPTIMAL SYNCHRONOUS BANDWIDTH ALLOCATION SCHEME FOR THE TIMED-TOKEN MAC PROTOCOL" PROCEEDINGS OF INFOCOM '95 - CONFERENCE ON COMPUTER COMMUNICATIONS. FOURTEENTH ANNUAL JOINT CONFERENCE OF THE IEEE COMPUTER AND COMMUNICATIONS SOCIETIES, BOSTON APR. 2 - 6, 1995, LOS ALAMITOS, IEEE COMP. SOC. PRESS, US, vol. 2 CONF. 14, 2 April 1995 (1995-04-02), pages 875-882, XP000580660 ISBN: 0-7803-2524-9	1,14
A	page 875, left-hand column, line 28 -right-hand column, line 30 page 876, left-hand column, line 46 -page 877, right-hand column, line 26	2-13, 15-26

—/—

[X] Further documents are listed in the continuation of box C.   [X] Patent family members are listed in annex.

° Special categories of cited documents :

'A' document defining the general state of the art which is not considered to be of particular relevance

'E' earlier document but published on or after the international filing date

'L' document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

'O' document referring to an oral disclosure, use, exhibition or other means

'P' document published prior to the international filing date but later than the priority date claimed

'T' later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

'X' document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

'Y' document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

'&' document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
14 April 2003	29/04/2003

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31–70) 340–2040, Tx. 31 651 epo nl, Fax: (+31–70) 340–3016	Lastoria, G

Form PCT/ISA/210 (second sheet) (July 1992)

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
Y	GENTER W L ET AL: "Delay analysis of the FDDI synchronous data class" MULTIPLE FACETS OF INTEGRATION. SAN FRANCISCO, JUNE 3 - 7, 1990, PROCEEDINGS OF THE ANNUAL JOINT CONFERENCE OF THE COMPUTER AND COMMUNICATIONS SOCIETIES (INFOCOM), WASHINGTON, IEEE COMP. SOC. PRESS, US, vol. 2 CONF. 9, 3 June 1990 (1990-06-03), pages 766-773, XP010019457 ISBN: 0-8186-2049-8	1,14
A	abstract page 766, left-hand column, line 1 -page 767, right-hand column, line 68; figure 1	1-13, 15-26
A	US 5 404 424 A (ZHAO WEI ET AL) 4 April 1995 (1995-04-04) column 4, line 6 -column 6, line 15 column 10, line 21 - line 34	1-26
A	SAHA D ET AL: "Carry-over round robin: a simple cell scheduling mechanism for ATM networks" PROCEEDINGS OF IEEE INFOCOM 1996. CONFERENCE ON COMPUTER COMMUNICATIONS. FIFTEENTH ANNUAL JOINT CONFERENCE OF THE IEEE COMPUTER AND COMMUNICATIONS SOCIETIES. NETWORKING THE NEXT GENERATION. SAN FRANCISCO, MAR. 24 - 28, 1996, PROCEEDINGS OF INFOCOM, L, vol. 2 CONF. 15, 24 March 1996 (1996-03-24), pages 630-637, XP010158124 ISBN: 0-8186-7293-5 page 632, right-hand column, line 1 -page 633, right-hand column, line 34	1-26
A	CHAN E ET AL: "ON THE EFFECTIVENESS OF THE FDDI-M MEDIUM ACCESS PROTOCOL FOR REAL-TIME TRAFFIC" JOURNAL OF SUPERCOMPUTING, KLUWER ACADEMIC PUBLISHERS, DORDRECHT, NL, vol. 11, no. 4, 1997, pages 371-389, XP000724664 ISSN: 0920-8542 page 371, line 5 -page 372, line 40 page 375, line 10 -page 377, line 11	1-26
P,A	WO 02 35777 A (MINGOZZI ENZO ;STEA GIOVANNI (IT); LENZINI LUCIANO (IT); SCARRONE) 2 May 2002 (2002-05-02) the whole document	1-26

Patent document cited in search report		Publication date	Patent family member(s)		Publication date
US 5404424	A	04-04-1995	NONE		
WO 0235777	A	02-05-2002	IT	T020001000 A1	23-04-2002
			AU	1519502 A	06-05-2002
			WO	0235777 A1	02-05-2002